---

title: What to Build description: Ideas for interesting things to build on SUAVE. keywords: - key - concepts - suave - build - ideas

---

What to Build on SUAVE

A good way to generate value is to design an app that is**uniquely enabled by SUAVE** ✨.

SUAVE exhibits three key differences from existing blockchains:

1. Programmable privacy
2. Credible offchain computation
3. Faster block time (than Eth L1)

Categories

In present-day Ethereum, there are many centralized services used to coordinate and mitigate the effects of MEV. Most of the categories below are based on these existing services:

- OFAs
- Searchers
- Solvers
- RFQ
- Intents
- AMMs
- Auctions
- PFOF
- Block Builders
- Rollup Scheduler
- Blob Merger

SUAVE also enables new and unique decentralized application paradigms:

- Social
- Gaming

OFA (Order Flow Auctions)

Order Flow Auctions (OFAs) are responsible for taking in a transaction or intent and routing it somewhere, either based on predefined logic, an auction, or by introspecting the transaction or intent.

You can learn more about it by reading"Illuminating Ethereum's Order Flow Landscape".

- MEV-Share
- MEV-Blocker
- MEV-Share+
- Dynamic refund: currently, users set their own refund percent, but depending on the state of the mempool and blockspace auction, they may want a lower or higher refund percent. A SUAPP could simulate a user's trade, query L1 state to get pool and token information, and use this to reason about what refund % to use on that user's bundle.
- Dynamic hint: one of the open research questions around programmable privacy is where the privacy efficiency lies. An example of this, relevant to MEV-Share, is as follows: given a Uniswap L1 trade, how much information should be revealed? A $10 trade most likely can leak the entire trade details because it would be better to batch, but a very large trade has a risk of being probabilistically front-run; therefore, it should leak less information, or at least that is one possibility.
- Batching user trades and back-runs together. In all these cases, a SUAPP could simulate a user's trade, query L1 state to get pool and token information, and use this to reason about what refund % to use on that user's bundle.

Searchers

Take an existing searcher bot design, write it in Solidity, and put it on-chain! Simple strategies will be more effective with the current state of the network and can take advantage of the sendBundle precompile to send bundles to centralized block builders.

Solvers

Solvers are responsible for finding mathematically optimal solutions in combinatorially large solution spaces, usually framed as finding optimal routes through multiple liquidity pools or in matching coincidences-of-wants. Solvers can be implemented

in Solidity and operate on order flow and intents submitted by users to SUAVE.

- CoWSwap Solver Example

RFQ (Request for Quote)

RFQs, or Requests for Quote, are off-chain services that take a user's ask, broadcast it to a network of market makers, and respond with the best price. A SUAPP could do this in two ways:

1. Use an HTTP post precompile to send to a list of market makers, or
2. Accept a curve of prices a market maker is willing to fill at and automatically respond on their behalf.

The second option changes the nature of RFQs, as market makers would not need to continuously update new pricing curves.

Intents

Intents refer to “what” the desired outcome of an action on a blockchain should be as opposed to transactions which specify “how” an action should be performed. Intents and Solvers work together very closely as some solvers aggregate intents in order to maximize user welfare under various definitions. In order to support these functionalities on SUAVE, you can help deploy various intent formats!

- ERC-7521
- ERC-4337: UserOps

AMMs (Automated Market Makers)

The ability for SUAVE to emit transactions and perform off-chain and confidential compute opens numerous doors for AMM designs.

- More expressive liquidity management
- Private liquidity
- UniV4 hook integrations

Auctions

Auctions, and mechanism design more broadly, can encompass most of the ideas on this page. But some specific interesting auction ideas include:

- Private NFT auction
- Combinatorial Auction
- Dutch Auctions - Fee escalators could be one example that are especially interesting combined with a programmable information leakage model like MEV-Share.
- Ad Auctions - Learn about header bidding first. Then, create a snippet/plugin that websites can integrate into their frontend so whenever a user visits the website, the data is sent to multiple auction houses/ad publishers, who then participate in an auction. In our case, the website owner would send the CCR to the auction contract, specifying some Kettle(s), and then the off-chain components do the actual auction and return the winning bid.

PFOF (Payment for Order Flow)

Design a SUAPP where wallets can forward order flow to and bulk auction off the transactions to searchers and market-makers.

Block Builders

There is a vast amount of builder algorithms live in the current Ethereum blockspace market; any of these can be implemented on SUAVE. Additional precompile support may be needed for more advanced functionality but can be merged relatively fast.

Rollup Scheduler

Rollups (L2s) have the challenge of posting data batches to L1, increased posting frequency increases security and UX but increases costs, less frequent posting is cheaper, but comes at the cost of UX. A Rollup post scheduler would be a SUAPP where each rollup submits their preference for when batches are uploaded and then stream batches to scheduler whenever ready. The scheduler will find optimal posting schedule for all preferences and post to base layer on each rollups behalf.

Blob Merger

In EIP 4844, each blob transaction can have a blob of a max size 128kb and blocks can have 4 blob txns. This means:

- given block with constant `BLOB_MAX_SIZE` = 128kb and `NUM_BLOBS_PER_BLOCK` = 2
- and 4 blob transactions w/ sizes: 100 kb, 100kb, 25kb, 25kb

we could never reach the optimal outcome (higher throughput) of all blobs being included. This is because the 100kb blobs will be leaving 28kb of space on the table, which the other two blobs could fit into. This is similar to the relation between the current gas limit and “regular” transactions in Ethereum, except in the data gas market case you’re only allowed two transactions and each transaction is storing temporary data, not initiating complex contract interaction. Luckily, EIP4844 Blob Transactions use BLS12-381 which can be easily aggregated. This is what a "Blob Merger" would do.

Social

Incentivize Twitter accounts to post certain tweets in specific time periods. Check results and post to chain with an oracle. Hide the content the account was incentivized to say using Confidential Data Storage .

Gaming

In games like Sid Meier’s Civilization, etc., you build an on-chain oracle that takes game state as input, and then you can allow players to sign (smart) contracts that let them negotiate with each other in a super expressive way, e.g., you can now say “I want to borrow your units on this continent for X turns and in exchange, I’ll commit to taking your future contracts of Y.” In this way, you can directly build mods (and specifically, mods around incentives/contracts/commitments) into existing games without having to build a new game, so you don’t have a cold-start problem and those mods are the most monetizable ones anyway .